

Update on the NAS Parallel Benchmarks

Rob F. Van der Wijngaart

Computer Sciences Corporation

NASA Ames Research Center

wijngaar@nas.nasa.gov

Collaborators:

Henry Jin, Michael Frumkin,

Parkson Wong, Huiyu Feng

All codes + information available at:

<http://www.nas.nasa.gov/Software/NPB>

Agenda

✍ Brief history of NAS Parallel Benchmarks (NPB)

✍ New NPB releases:

- larger problem sizes (Class D)
- (parallel) I/O
- new paradigms: Java, OpenMP, HPF

✍ Release of NAS Grid Benchmarks

✍ Future plans for benchmarking

- irregular, dynamically changing memory accesses:
Unstructured Adaptive-mesh UA
- multi-level parallel applications
- spruced up NPB web site
- data intensive applications

Brief NPB Release History

- ✍ 1991: Paper-and-pencil specification (NPB 1)
- ✍ 1995: MPI implementation, classes S, A, B (NPB2.0)
- ✍ 1996: MPI implementation, classes W, C (NPB2.2)
- ✍ 2002:
 - MPI implementation, class D (NPB2.4)
 - BTIO paper-and-pencil specification + MPI implementation (NPB2.4)
 - HPF, JAVA, OpenMP implementations (NPB3.0)
 - Grid Benchmark paper-and-pencil specification (NGB 1)
 - Non-distributed NGB implementation (GridNPB3.0)
- ✍ 2003: Distributed JAVA NGB implementation (GridNPB3.0)

Scaling up NAS Parallel Benchmarks

- ✍ No new classes introduced since 1996 (C)
- ✍ System sizes increased: ~2.5
- ✍ Processor power increased substantially: ~20
- ✍ Caches increased substantially: ~10-30
- ✍ Top 500, 5th place:
 - Nov 1996- Intel XP: 127 GFlops/3000 procs
 - Nov 2001- IBM SP: 2144 GFlops/4000 procs

Scaling up NAS Parallel Benchmarks

Scaling rationale:

- ✍ Largest problem class should run in approximately constant time at any one time
- ✍ Cache size/data size for largest class should remain roughly constant
- ✍ Class B released 1992, class C released 1996
Plan: release new class every 4 years
 - Class D overdue
 - Class E a bit premature: extrapolate

Class D scaling approach:

- ✍ System power increased ~ 20
- ✍ Problem size = iterations x data size
- ✍ Increase iterations by 1.25, data size by 16

New Class D Problem Sizes

	MG	CG	FT	SP, BT, LU
C	512^3	150,000/15	512^3	162^3
D	1024^3	1,500,000/21	2048×1024^2	408^3

[http://www.nas.nasa.gov/Research/Reports: NAS-02-007](http://www.nas.nasa.gov/Research/Reports/NAS-02-007)

MPI implementation issues (Rob Van der Wijngaart):

- ✍ Class D problems already require 64 bits on small numbers of procs
- ✍ Some (FT,IS) require large integers: MPI API trouble

Numerical issues: convergence (CG)

Parallel I/O

Rationale:

- ✍ NPBs do no I/O to speak of
- ✍ Scientific codes that matter produce/consume data commensurate with amount of computation

Typical types of I/O:

- ✍ Checkpointing/post-processing
 - typically only output
 - don't want domain decomposition reflected in structure of stored data (persistent)
- ✍ Out-of-core solvers
 - both output and input
 - don't care about structure of data on disk (temporary)

Parallel I/O

NPB2.4-IO (formerly BTIO, Parkson Wong):

- ✍ Checkpoints BT solution array every 5 time steps
- ✍ Significance: may overlap I/O and computation
- ✍ Verifies final and all intermediate results upon completion of computation (verification not timed)
- ✍ Output sizes in GB: 0.42 (A), 1.7 (B), 6.8 (C), 136 (D)
- ✍ Reference parallel implementation:
 - Uses MPI, multi-partition domain decomposition
 - Four different output modes:
 - ✍ Full: MPI-IO with collective buffering
 - ✍ Simple: MPI-IO without collective buffering
 - ✍ Fortran: Plain Fortran I/O
 - ✍ EPIO: separate output files for each processor (cheating!)

<http://www.nas.nasa.gov/Research/Reports>: NAS-03-002

New paradigms

NPB3.0 (formerly PBN, Henry Jin, Michael Frumkin):

- ✍ Derived from improved serial version of NPB2.3
- ✍ OpenMP, Java, High Performance Fortran

`http://www.nas.nasa.gov/Research/Reports:`

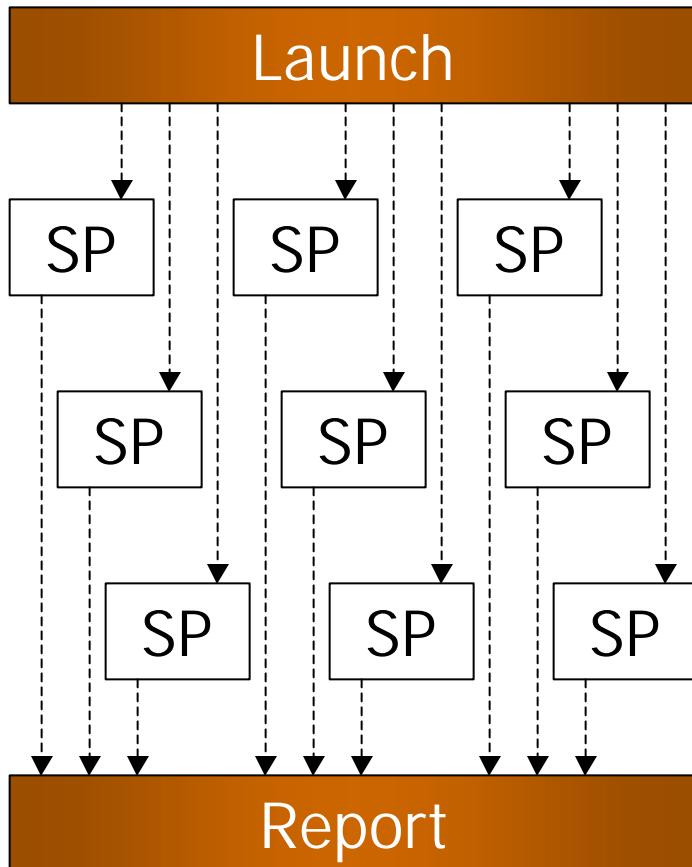
- ✍ NAS-98-009 (High Performance Fortran)
- ✍ NAS-99-011 (OpenMP)
- ✍ NAS-02-009 (Java)

NAS Grid Benchmarks (NGB)

- ✍ Gauge grid performance for (distributed) scientific applications
- ✍ 4 compound tasks: 3 pseudo apps, 2 kernels, all derived from NPB
- ✍ Measure turnaround time (includes time in queue)
- ✍ Paper-and-pencil specifications: NGB1
[http://www.nas.nasa.gov/Research/Reports: NAS-02-005](http://www.nas.nasa.gov/Research/Reports/NAS-02-005)
(also proposed to as Global Grid Forum standard)
- ✍ Source code implementations: GridNPB3.0
 - Non-distributed, single file system (Rob Van der Wijngaart)
 - Distributed Java, fully concurrent (Michael Frumkin)

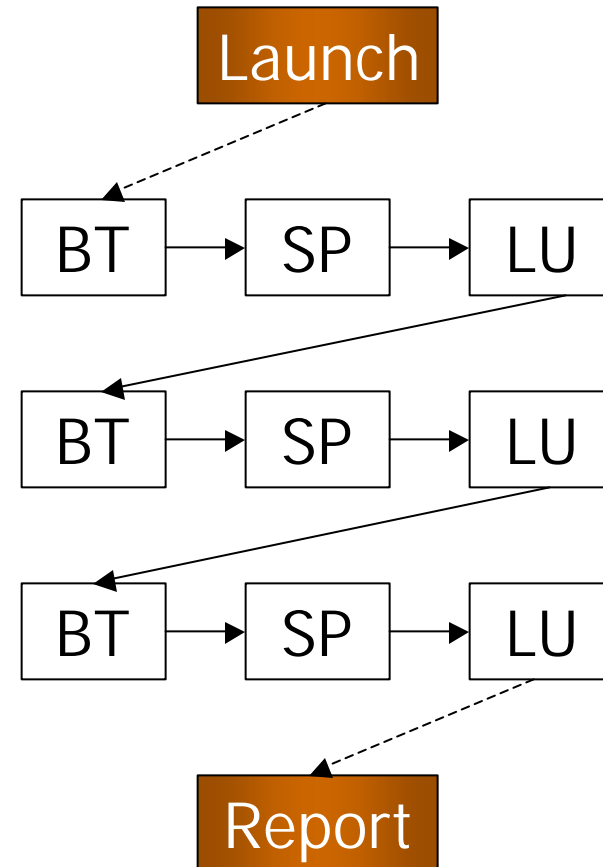
NGB Data Flow Graphs, Sample Size

Embarrassingly Distributed (ED)



Parameter study

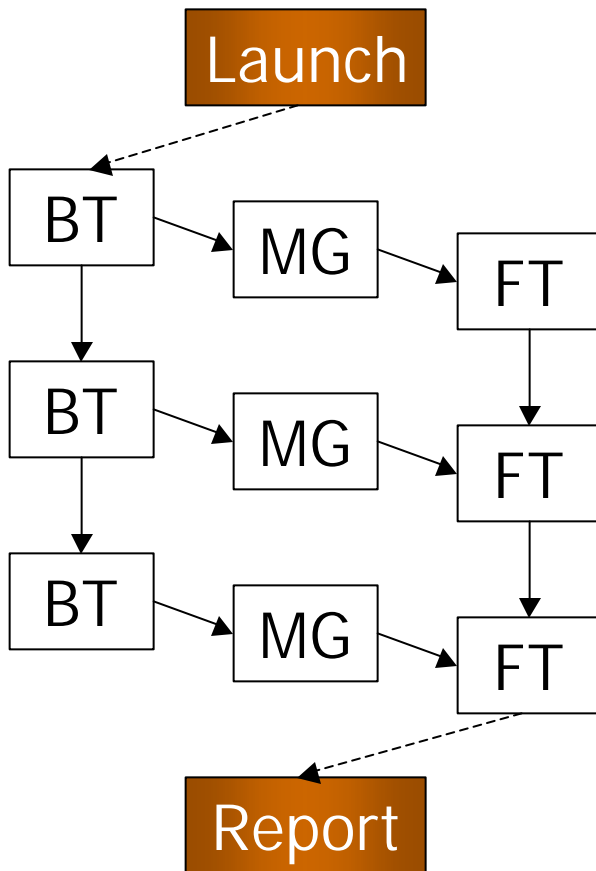
Helical Chain (HC)



Cyclic process (restart)

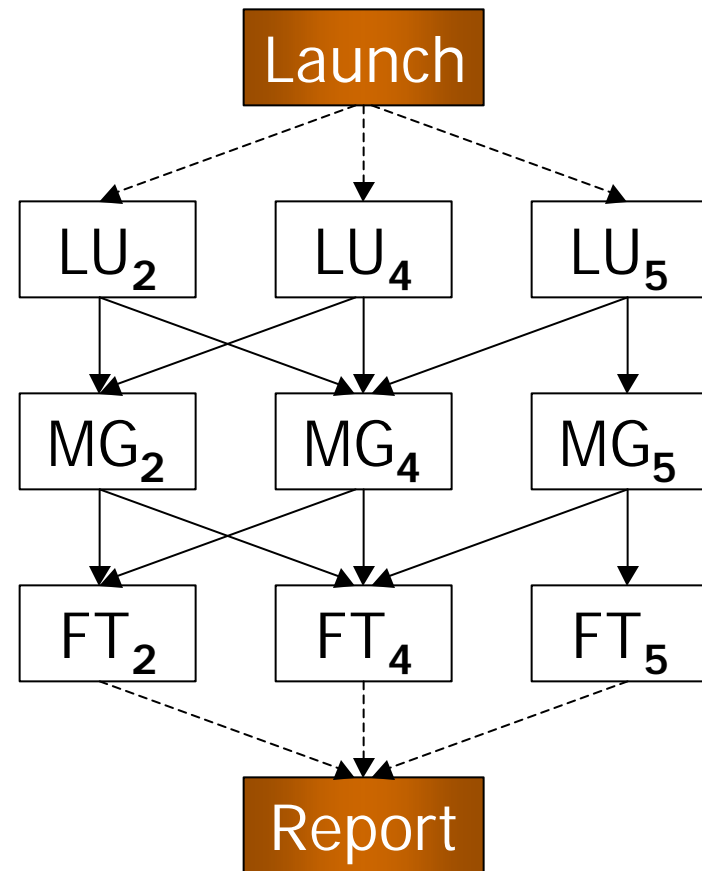
NGB Data Flow Graphs, Sample Size

Visualization Pipe (VP)



Visualization cycle

Mixed Bag (MB)



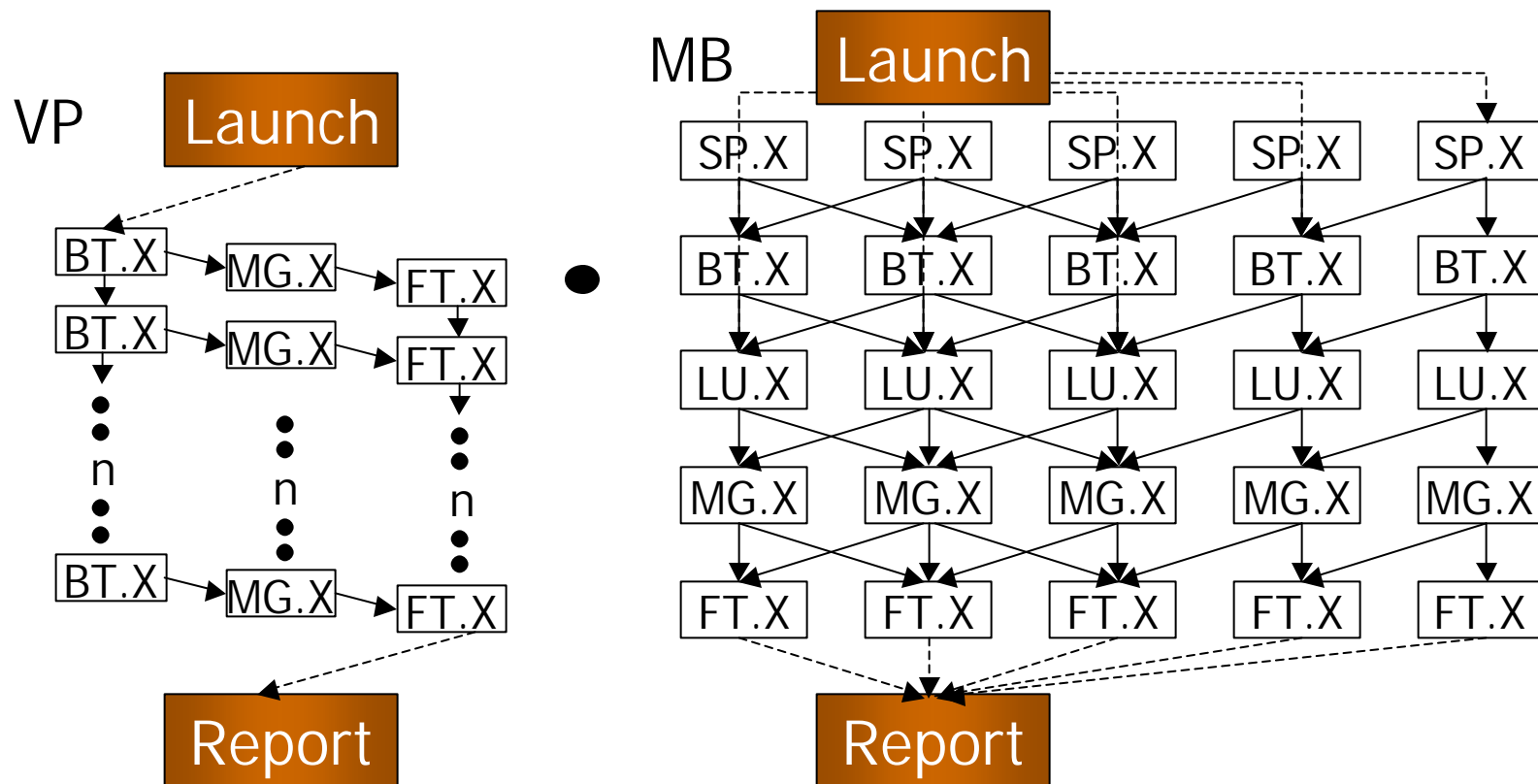
Unbalanced chain

Larger NGB Problem Sizes





$(ED.X \mid X? \{A,B,C\}) = ? \{9,18,36\} SP.X$

$(HC.X \mid X? \{A,B,C\}) = ? \{3,6,9\} (BT.X, SP.X, LU.X)$



$(VP.X \mid X? \{A,B,C\}) = ? \{3,6,9\} (BT.X, MG.X, FT.X)$



Unstructured Adaptive-mesh UA

-  Simple 3D heat transfer problem
-  Prescribed motion of heat source (= error source)
-  Adapt mesh periodically where needed
-  Numerical approach:
 - Non-conforming spectral finite elements (rectangular)
 - Explicit integration of convection term
 - Implicit integration of diffusion term;
solve linear system using PCG

Implementation (Huiyu Feng):

-  Serial version ready, being tested
-  MPI version expected ready by end of 2003

Multi-level Parallel Benchmarks

Many apps have several levels of exploitable parallelism:






- ✍ Deeply nested loops (already in NPBs)
- ✍ Multiple loosely-coupled discretization grids ✍ hybrid codes

Approach: Derive problems from existing NPBs: LU, BT, SP

- Divide original grids into 2D sets of overlapping grids
- Initialize solutions as in single-grid case
- Exchange bdry values after each time step (periodic in horizontal)
- LU-MB: fixed number of equal-sized grids (4x4x1)
- SP-MB: growing number of equal-sized grids (NxNx1)
- BT-MB: growing number of unequal-sized grids (NxNx1)


Expected ready by summer 2003 (Rob Van der Wijngaart)


Spruce up NPB web site

-  Accept new results for all NPBs, including GridNPB
-  Online submission of results
-  Support for many different types of results database queries
-  On-the-fly plotting of data
-  Expected ready by end of summer 2003

Data Intensive (Grid) Benchmarks

Computational intensity:

 a posteriori:
$$\frac{\text{time spent in comps}}{\text{time spent doing I/O}} \quad \frac{\# \text{ computational ops}}{\# \text{ I/O ops}}$$

 a priori:
$$\frac{\# \text{ computational ops}}{\# \text{ referenced elms of data set}}$$

Data intensity:
$$\frac{1}{\text{computational intensity}}$$

Issues:

 paper-and-pencil specs \longrightarrow a priori intensities

 must specify

- I/O operations
- computational operations
- data set